BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

98P7347

(4)

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号

B4

特開平9－298537

(43)公開日　平成９年(1997)11月18日

(51)Int.Cl.[6]		識別記号	庁内整理番号	Ｆ Ｉ		技術表示箇所
Ｈ０４Ｌ	9/32			Ｈ０４Ｌ	9/00	６７５Ｂ
Ｇ０９Ｃ	1/00	６４０	7259－5Ｊ	Ｇ０９Ｃ	1/00	６４０Ｂ

審査請求　未請求　請求項の数13　ＯＬ　（全 13 頁）

(21)出願番号　特願平8－108226

(22)出願日　平成８年(1996)４月26日

(71)出願人　000001007
キヤノン株式会社
東京都大田区下丸子３丁目30番２号

(72)発明者　大石　和臣
東京都大田区下丸子３丁目30番２号　キヤノン株式会社内

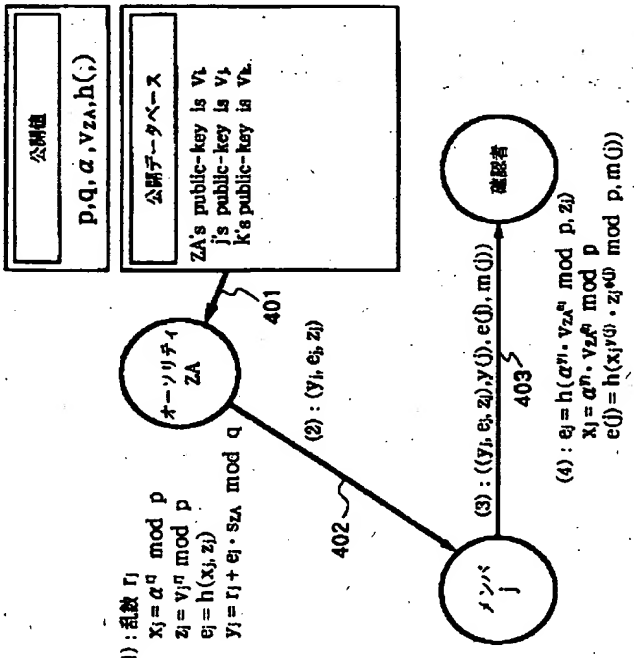
(74)代理人　弁理士　大塚　康徳　（外１名）

(54)【発明の名称】　ディジタル署名方式およびそれを用いた情報通信システム

(57)【要約】

【課題】　グループ署名の正当性を確認するために用いるグループの公開鍵は、メンバの数に比例した大きさになる。

【解決手段】　オーソリティZAは、メンバjに対してメンバであることの証明書であるディジタル署名を送る。証明書を入手したメンバjは、署名したいメッセージm(j)に対し、グループ署名を生成する。メッセージm(j)を受信した確認者は、そのグループ署名が、オーソリティZAに認められたメンバによって生成されたグループ署名か否かを確認する。



1

【特許請求の範囲】

【請求項１】　グループメンバに用いられ、ディジタル署名を生成する生成手段と、

グループの公開鍵を用いて、対象とする署名が前記グループメンバにより生成されたことを確認する確認手段と、

対象とする署名の署名者を開示する開示手段とを有することを特徴とするディジタル署名方式。

【請求項２】　前記グループの公開鍵は、グループメンバ数に比例しない大きさであることを特徴とする請求項1に記載されたディジタル署名方式。

【請求項３】　前記確認手段は、対象とする署名の署名者を確認できないことを特徴とする請求項1または請求項2に記載されたディジタル署名方式。

【請求項４】　前記開示手段は、前記署名者を開示するために特別なユーザを用いないことを特徴とする請求項1に記載されたディジタル署名方式。

【請求項５】　前記開示手段による前記署名者の開示は、特別なユーザを用いる場合と、前記特別なユーザを用いない場合とがあることを特徴とする請求項1に記載されたディジタル署名方式。

【請求項６】　複数ユーザ間で共通に使われる第一の底の値に対して各ユーザの秘密鍵を指数値として指数計算を行った結果を各ユーザの第一の公開鍵とするディジタル署名方式であって、

前記底に対して乱数を指数値として指数計算を行った結果を第二の底とする底生成ステップと、

所定のユーザの公開鍵に対して前記乱数を指数値として指数計算を行った結果を第二の公開鍵とする公開鍵生成ステップと、

前記所定のユーザが、自身の秘密鍵を用いて、任意の平文に対して第一のディジタル署名を生成する第一の署名生成ステップと、

前記第二の底および前記第二の公開鍵を用いて、前記第一のディジタル署名の正当性を確認する確認ステップと、

単一または複数の特別なユーザが、前記複数のユーザそれぞれに対応する前記第二の公開鍵に対して、第二のディジタル署名を生成する第二の署名生成ステップと、

前記第一および第二のディジタル署名の組からなる第三のディジタル署名から、その署名者を開示する開示ステップとを有することを特徴とするディジタル署名方式。

【請求項７】　離散対数を求めることの困難性に安全性の根拠をおくことを特徴とする請求項6に記載されたディジタル署名方式。

【請求項８】　Schnorr署名方式および/またはElGamal署名方式および/またはDSA方式を用いることにより、前記離散対数を求めることを困難にすることを特徴とする請求項7に記載されたディジタル署名方式。

【請求項９】　前記開示ステップは、前記単一または複数の特別なユーザにより署名者を開示することを特徴とする請求項6から請求項8の何れかに記載されたディジタル署名方式。

【請求項１０】　前記開示ステップは、前記グループメンバそれぞれが、前記第一のディジタル署名の生成者が自分であるか否かを証明することを特徴とする請求項6から請求項8の何れかに記載されたディジタル署名方式。

【請求項１１】　請求項1から請求項10の何れかに記載されたディジタル署名方式を用いることを特徴とする情報通信システム。

【請求項１２】　ディジタル署名方式に関するプログラムコードが格納されたコンピュータ可読メモリであって、

グループメンバに用いられ、ディジタル署名を生成する生成手段のコードと、

グループの公開鍵を用いて、対象とする署名が前記グループメンバにより生成されたことを確認する確認手段のコードと、

対象とする署名の署名者を開示する開示手段のコードとを有することを特徴とするコンピュータ可読メモリ。

【請求項１３】　複数ユーザ間で共通に使われる第一の底の値に対して各ユーザの秘密鍵を指数値として指数計算を行った結果を各ユーザの第一の公開鍵とするディジタル署名方式を用いた情報通信システムに関するプログラムコードが格納されたコンピュータ可読メモリであって、

前記底に対して乱数を指数値として指数計算を行った結果を第二の底とする底生成ステップのコードと、

所定のユーザの公開鍵に対して前記乱数を指数値として指数計算を行った結果を第二の公開鍵とする公開鍵生成ステップのコードと、

前記所定のユーザが、自身の秘密鍵を用いて、任意の平文に対して第一のディジタル署名を生成する第一の署名生成ステップのコードと、

前記第二の底および前記第二の公開鍵を用いて、前記第一のディジタル署名の正当性を確認する確認ステップのコードと、

単一または複数の特別なユーザが、前記複数のユーザそれぞれに対応する前記第二の公開鍵に対して、第二のディジタル署名を生成する第二の署名生成ステップのコードと、

前記第一および第二のディジタル署名の組からなる第三のディジタル署名から、その署名者を開示する開示ステップのコードとを有することを特徴とするコンピュータ可読メモリ。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明はディジタル署名方式およびそれを用いた情報通信システムに関する。

3

【0002】

【従来の技術】コンピュータおよび通信ネットワークの発展と広範な普及に伴い、社会的活動のような機能をネットワーク上においても実現できるようになった。しかし、誰が、何時、何処で、何を行ったか、も容易に把握することができ、それに対抗するため、匿名で処理を行うことによりネットワーク上で多様な機能を実現し、かつプライバシも保護する方法が考えられている。

【0003】後述する公開鍵暗号を用いれば、送信者は通信内容を意図する受信者だけに送り、しかも受信者はその通信の送信者が誰であるかを確実に確認することが可能である。後述する零知識証明を利用すれば、ユーザがある秘密情報を保持していることを、その秘密を明らかにすることなく他者に証明できる。これらを応用した情報通信システムが、Masahiro MAMBO, Eiji OKAMOTO, "A method to publicly specify a signer with hiding identity, The 18th Symposium on Information Theory and Its Applications" (October 1995)で提案されている。これを、以下では「MOシステム」と呼ぶことにする。

【0004】以下、最初に公開鍵暗号を、次に零知識証明を、それからMOシステムを詳細に説明する。

【0005】［公開鍵暗号］公開鍵暗号とは、暗号鍵と復号鍵とが異なり、暗号鍵を公開し、復号鍵を秘密にする暗号方式であり、以下のような特徴をもつ。

(1)暗号鍵と復号鍵とが異なり、暗号鍵を公開することができるため、暗号鍵を秘密に配送する必要がなく、鍵の配送が容易である。

(2)各利用者は、暗号鍵を公開し、各自、復号鍵だけを秘密にし記憶しておけばよい。

(3)送られてきた通信文の送信者が偽者でないこと、および、その通信文が改竄されていないことを、受信者が確認するための認証機能（ディジタル署名）を実現できる。

【0006】通信文（平文）Mに対して、公開の暗号鍵kpを用いた暗号化操作をE(kp, M)とし、秘密の復号鍵ksを用いた復号操作をD(ks, M)とすると、公開鍵暗号アルゴリズムは、まず次の二つの条件を満たす。

(1)暗号鍵kpが与えられたときE(kp, M)の計算は容易であり、復号鍵ksが与えられたときD(ks, M)の計算は容易である。

(2)もし復号鍵ksを知らないなら、暗号鍵kp、暗号化操作Eの計算手順、暗号であるC=E(kp, M)を知ったとしても、平文Mを決定することは、その計算量から困難である。

【0007】さらに、上記(1)(2)に加えて、次の条件(3)が成立することにより、秘密通信が実現できる。

(3)すべての平文Mに対し、E(kp, M)を定義することができ、次式が成立する。

D(ks, E(kp, M)) = M

4

【0008】つまり、暗号鍵kpは公開されているため、誰もがE(kp, M)を計算することができるが、D(ks, E(kp, M))を計算して平文Mを得ることができるのは、秘密の復号鍵ksを記憶している本人だけである。

【0009】一方、上記(1)(2)に加えて、次の条件(4)が成立することにより、認証通信（ディジタル署名）が実現できる。

(4)すべての平文Mに対し、D(ks, M)を定義することができ、次式が成立する。

E(kp, D(ks, M)) = M

【0010】つまり、D(ks, M)を計算できるのは秘密の復号鍵ksを記憶している本人だけであり、他人が偽の鍵ks'を用いてD(ks', M)を計算し、秘密の復号鍵ksを記憶する本人になりすまそうとしても、E(kp, D(ks', M))≠Mであり、受信者は受取った情報が不正なものであることを知ることができる。

【0011】同様に、D(ks, M)が改竄されても、E(kp, D(ks, M)')≠Mであり、受信者は受取った情報が不正なものであることを知ることができる。このD(ks, M)をMに対する「ディジタル署名」と呼ぶ。

【0012】代表的な公開鍵暗号方式を以下に挙げる。秘密通信と認証通信ができる方式として、RSA暗号(R. L. Rivest, A. Shamir and L. Adleman: "A method of obtaining digital signatures and public key cryptosystems", Comm. of ACM, 1978)、R暗号(M. Rabin: "Digitalized signatures and public-key cryptosystems", MIT/LCS/TR-212, Technical Report MIT, 1979)、W暗号(H. C. Williams: "Amodification of the RSA public-key encryption procedure", IEEE Trans. Inf. Theory, IT-26, 6, 1980)などがある。

【0013】また、秘密通信だけができる方式として、CR暗号(B. Chor and R. L. Rivest:"A knapsack type public key cryptosystem based on arithmetric in finitefield", Proc. Crypto 84)、M暗号(R. J. McEliece: "A public-key cryptosystem based on algebraic coding theory", DSN Progress Rep., Jet PropulsionLab., 1978)、E暗号(T. E. ElGamal: "A public key cryptosystem and a signature scheme based on discretre logarithms", IEEE Transaction on Information Theory, Vol. IT-31, No.4, pp.469-472, 1985)などがある。

【0014】認証通信だけができる方式として、S暗号(A. Shamir: "A fast signature scheme", Report MIT/LCS/TM-107, MIT laboratory for computer science, Cambridge, Mass., 1978)、L暗号(K. Lieberherr: "Uniform complexity and digitalsignature", Lecture Notes in Computer Science 115 Automata, Language andProgramming, Eigtht Colloquium Acre, Israel, 1981)、GMY暗号(S. Goldwasser, S. Micali and A. Yao: "Strong signature schemes", ACM Symp. on Theory of Comp

5

uting, 1983)、GMR暗号(S. Goldwasser, S. Micali and R. L. Rivest: "A 'paradoxical' solution to the signature problem", ACM Symp. on Foundation of Computer Science, 1984)、E暗号(T. E. ElGamal: "A public key cryptosystem and a signature scheme based on discretre logarithms", IEEE Transaction on Information Theory, Vol. IT-31, No.4, pp.469-472, 1985)、OS暗号(岡本, 白石: "多項式演算によるディジタル署名方式", 信学論(D), J68－D, 5, 1985; T. Okamoto and A. Shiraisi: "A fast signature scheme based on quadratic inequalities", IEEE Symp. on Theory of Computing, 1984)、

【００１５】Fiat-Shamir暗号(A. Fiat, A. Shamir: "How to prove yourself: practicalsolutions of identification and signature problems", Proc. of CRYPTO'86,1987)、Schnorr暗号(C. P. Schnorr: "Efficient signature generation by smart cards", Journal of Cryptology, vol.4, pp.161-174, 1991)などがある。

【００１６】具体的な一例として、離散対数を求めることの困難性に安全性の根拠をおき、偽造に対する安全性が極めて高いとされ、署名のサイズが小さい署名方式であるSchnorr暗号を説明する。なお、以下の説明において、a^bはaのb乗を表す。

【００１７】例えば、大きな素数pと、q|p-1（qはp-1を割り切る）なる大きな素数qを選ぶとき、{1, 2, …, q-1}の中の一つをαとすると、α^x≡u (mod p)を満たすx (=DL(u,α)と記す）は、ガロア体GF(p)上における離散対数であり、p, q, α, uを与えられた上でxを効率的に求めることは難しいとされている。

【００１８】Schnorr暗号では、鍵認証センタ(KAC: Key Authentication Center)がシステムの初期化を行う。署名者が鍵生成と署名生成を行い、認証者が署名を受取り確認する。システムの中では、一方向性ハッシュ関数を用いる。一方向性ハッシュ関数とは、衝突を起こしにくい圧縮関数である。つまり、任意の長さの値を入力して固定の長さの値を出力し、同じ値を出力する異なる入力を見つけることが困難である。Zは整数の集合で、Zqは0以上q未満の整数の集合である。

【００１９】●システム初期化

(1)鍵認証センタ(KAC)は、大きな素数p(≧2^512)とq(≧2^140)を選び公開する。ただし、q|p-1である。

(2)KACは、α∈Zpかつα^q = 1 (mod p)なるα(≠1)を選び公開する。

(3)KACは、h: Zq×Z → {0, …, 2^t - 1}を選び公開する。ここで、tを「セキュリティパラメータ」と呼ぶ。

(4)KACは、自分の署名用の公開鍵と秘密鍵を選び、公開鍵を公開する（例えば、Fiat-Shamir暗号など）。

【００２０】●鍵生成

システムに加入するユーザ（署名者）は、0 < s≦qなる任意の整数sを選び、v= α^{-s} mod pを計算する。署

6

名者の秘密鍵はsであり、公開鍵はvである。認証者は、署名者の公開鍵の正当性を、例えば、署名者の公開鍵に対してKACが（例えば、Fiat-Shamir暗号で）署名を付ける、あるいは、署名者の公開鍵が公開データベースに登録される、などによって確認することができる。

【００２１】●署名生成（メッセージmに対する署名）

(1)署名者は、乱数r∈{1, …, q}を選び、x = α^r (mod p)を計算する。

(2)署名者は、e = h(x, m)を求める。

(3)署名者は、y = r + s･e (mod q)を計算する。(e, y)が、メッセージmに対する署名である。

【００２２】●署名検証

メッセージmと署名(e, y)を受取った認証者は、前述の方法でvの正当性を確認し、次式を確認する。

x' = α^y･v^e (mod p)

e = h(x', m)

【００２３】［零知識証明］零知識証明とは、ある命題を証明者が検証者に証明することであり、以下の三条件を満たす。

(1)完全性: 証明が正しければ、検証者は1または1に限りなく近い確率（圧倒的確率）で受理する。

(2)健全性: 証明が誤っていれば、検証者は1または1に限りなく近い確率（圧倒的確率）で棄却する。

(3)零知識性: 証明が正しければ、検証者がどのように振る舞っても、証明の正しさ以外の情報は一切洩れない。

【００２４】ある秘密（数値）を知っている証明者が、その秘密に関する情報を一切漏らさずに、その秘密を知っていることを検証者に証明する零知識証明の方法が幾つか提案されていて、身元証明やディジタル署名などに応用され、情報セキュリティの分野における重要な基板技術となっている。

【００２５】具体的な一例として、J. Boyar, D. Chaum, I. Damard, T. Pedersen, "Convertible undeniable signatures", Proc. of CRYPTO'90において提案された、z =DL(u, α)を知っている証明者が、zを明らかにすることなくDL(v, w) = DL(u,α)を証明する零知識証明プロトコルを説明する。ただし、DL(u, α)は適当な群における離散対数を意味する（例えば、前述のGF(p)の場合、α^{DL(u, α)} ≡(mod p)）。

【００２６】●DL(v, w) = DL(u, α)を証明する零知識証明プロトコル

以下では、証明者をP(Prover)、検証者をV(Verifier)と記す。PもVもv, w, u, αを知っている。

(1)検証者Vは、乱数aとbを選び(a, b∈Zq)、ch = w^a･α^bを求め、chを証明者Pに送る。

(2)証明者Pは、乱数tを選び(t∈Zq)、h1 = ch･α^tとh2 = h1^zを求め、(h1, h2)を検証者Vに送る。

(3)検証者Vは、(a, b)を証明者Pに送る。

(4)証明者Pは、ch = w^a･α^bを確認し、tを検証者Vに

送る。
(5)検証者Vは、h1 = w^a·α^(b+t)とh2 = v^a·u^(b+t)を確認する。
【００２７】［ディジタル署名方式］以上に説明した公開鍵暗号や零知識証明を応用したディジタル署名方式が、D.Chaum, E. van Heyst, "Group Signatures", Proc. of EUROCRYPT'91, pp.257-265 (1991)に提案されている。これはグループ署名と呼ばれ、以下の性質を満たす。
(1)グループメンバのみが署名を行える。
(2)署名を受取った者は、それが正当な署名であることは確認できるが、グループのどのメンバが証明をしたのかはわからない。
(3)後で必要が生じたときに署名者を明らかにするために、署名は（グループメンバの助力を得て、あるいは、得ずに）開示されることができる。
【００２８】グループ署名の用途には、例えば、入札システムがある。ある入札に参加する応札者をメンバとし、応札者は自分の付ける値段や条件に対してグループ署名を生成する。落札しない限り誰がどんな値段や条件を提示したかは不明であるが、落札の際には落札者、つまり署名者を明らかにすることができる。
【００２９】
【発明が解決しようとする課題】前述したChaum等の論文には四つのグループ署名方式が提案されていて、それぞれ、署名者名を開示する際に特別に信頼されたオーソリティ（以下、これをZAで表す）を要する方式と、そうでない方式、グループメンバの新規加入が容易な方式と、そうでない方式など、幾つかの基準により分類される特徴をもつ。しかし、その何れの方式も、署名の正当性を確認するために用いるグループの公開鍵は、メンバの数に比例した大きさになるという性質がある。
【００３０】一方、S. J. Park, I. S. Lee, D. H. Won, "A practical group signature",Proc. of the 1995 Japan-Korea Workshop on Information Security and Cryptology, IV-3, pp.127-133 (1995)に提案されているグループ署名方式は、グループの公開鍵の大きさはメンバ数に比例せず、メンバの新規加入が容易であるという特徴をもち、署名の開示の際にZAが必要な方式である。
【００３１】本発明は、上述の問題を解決するためのものであり、グループの公開鍵の大きさがグループメンバ数に比例する必要がないディジタル署名方式およびそれを用いた情報通信システムを提供することを目的とする。
【００３２】また、メンバの新規加入が容易であるディジタル署名方式およびそれを用いた情報通信システムを提供することを他の目的とする。
【００３３】また、オーソリティが署名の開示を行うことが可能であるとともに、各メンバが、その署名は自分が作成した署名か否かを証明することが可能なディジタル署名方式およびそれを用いた情報通信システムを提供することを他の目的とする。
【００３４】
【課題を解決するための手段】本発明は、前記の目的を達成する一手段として、以下の構成を備える。
【００３５】本発明にかかるディジタル署名方式は、グループメンバに用いられ、ディジタル署名を生成する生成手段と、グループの公開鍵を用いて、対象とする署名が前記グループメンバにより生成されたことを確認する確認手段と、対象とする署名の署名者を開示する開示手段とを有することを特徴とする。
【００３６】また、複数ユーザ間で共通に使われる第一の底の値に対して各ユーザの秘密鍵を指数値として指数計算を行った結果を各ユーザの第一の公開鍵とするディジタル署名方式であって、前記底に対して乱数を指数値として指数計算を行った結果を第二の底とする底生成ステップと、所定のユーザの公開鍵に対して前記乱数を指数値として指数計算を行った結果を第二の公開鍵とする公開鍵生成ステップと、前記所定のユーザが、自身の秘密鍵を用いて、任意の平文に対して第一のディジタル署名を生成する第一の署名生成ステップと、前記第二の底および前記第二の公開鍵を用いて、前記第一のディジタル署名の正当性を確認する確認ステップと、単一または複数の特別なユーザが、前記複数のユーザそれぞれに対応する前記第二の公開鍵に対して、第二のディジタル署名を生成する第二の署名生成ステップと、前記第一および第二のディジタル署名の組からなる第三のディジタル署名から、その署名者を開示する開示ステップとを有することを特徴とする。
【００３７】本発明にかかる情報通信システムは、グループメンバに用いられ、ディジタル署名を生成する生成手段と、グループの公開鍵を用いて、対象とする署名が前記グループメンバにより生成されたことを確認する確認手段と、対象とする署名の署名者を開示する開示手段とを有するディジタル署名方式を用いることを特徴とする。
【００３８】
【発明の実施の形態】以下、本発明にかかる一実施形態の情報通信システムを図面を参照して詳細に説明する。
【００３９】本発明にかかるディジタル署名方式は、後述する匿名公開鍵証明書方式を応用して、信頼されたオーソリティZAが指定者、メンバが署名者になり、匿名公開鍵証明書に基づくディジタル署名をグループ署名とするものであり、本発明にかかる匿名公開鍵証明方式によれば、以下のような機能を実現することができる。
【００４０】指定者が、乱数により、共通の公開パラメータと署名者のディジタル署名の公開鍵を変換し、変換した公開鍵に対する指定者のディジタル署名1を生成する。署名者の秘密鍵と、変換されたパラメータおよび公

開鍵との間には、以前と同じ関係が成り立つので、署名者は秘密鍵を用いて任意の平文に対してディジタル署名2を生成することができる。

【００４１】ディジタル署名1および同2の受信者は、ディジタル署名1により、変換されたパラメータおよび公開鍵が指定者に認められていることを確認し、それらを用いて、ディジタル署名2と平文の対応関係を確認することができる。なお、ディジタル署名1および同2を合わせた署名は、それから署名者を明らかにすることが困難なので匿名署名と呼ぶ。

【００４２】このように、指定者が信頼されたオーソリティZAになり、指定者に指定されたユーザ、すなわち署名者がメンバになるので、オーソリティZAの使う公開鍵はグループの公開鍵になり、匿名署名はグループ署名になる。この署名はグループ署名の最初の二つの条件を満たす。

【００４３】署名者の開示は、次のような方法をとる。信頼されたオーソリティZAは、どのメンバにどの証明書を発行したかを記録し、グループ署名が与えられたとき、その証明書の部分からそれがどのメンバにより生成されたかを判別することができる。さらに、その証明書に用いた乱数を知っているので、その乱数を利用して証明者名を開示することができる。

【００４４】一方、あるグループ署名に対して、各メンバがその署名者であるか否かは、そのグループ署名に使われた秘密鍵と各メンバの秘密鍵が等しいか否かで判別することができる。従って、署名者ではないメンバは、自分の秘密鍵と対象のグループ署名に使われた秘密鍵とが異なることを証明できるので、すべてのメンバに対して、その証明を要求することにより誰が証明者であるかを判別することができる。

【００４５】

【第1実施形態】以下では、匿名公開鍵証明書方式(APKC: Anonymous Public Key Certificate)を最初に説明し、次に、それを応用したグループ署名を説明する。

【００４６】［匿名公開鍵証明書方式］図1は公開鍵暗号を応用した匿名公開鍵証明書を説明するための図で、図2は公開鍵証明書に関する処理の流れの概要を示すフローチャートである。

【００４７】このシステムは、指定者（証明書の発行者）と複数のユーザ、ユーザの中から選ばれる署名者、署名者の作成する署名を確認する確認者からなり、通信内容から判別できる場合を除き送受信者を突き止めることが困難な通信を可能とするネットワーク（以下「匿名＊通信ネットワーク」と呼ぶ）上で実現することができる。

【００４８】図1において、「公開値(Public Values)」は後述するシステム共通のデータ、「公開データベース(Public Database)」は公開されたデータベース、矢印はデータの取得，送信，受信を示し、括弧で囲まれた番号は処理の手順を示す。また、「署名者(Specified Signer)」は、「指定者(Specifier)」によりユーザの中から選ばれた署名者を表し、「確認者(Verifier)」は署名を確認する者を表す。

【００４９】Step 0（準備）：システム共通のデータとして、素数pとq(q|p-1)、Zp*の元であり位数qのα（α^q≡1 (mod p)、Zp*はZqかつpと互いに素である整数の集合）、一方向性ハッシュ関数h: Zq×Z → {0, …, 2^t - 1}を用意する。これらは、システムに参加しているすべてのユーザがアクセスでき、かつ不当な改竄などが起こらないように適切に管理されている公開のデータベースに登録されているものとする。なお、以下の説明において、a_bはaに下付き添字bが付いた状態を表す。

【００５０】指定者iは公開鍵v_iと秘密鍵s_i(v_i = α^{-s_i} mod p)を生成し、公開鍵を公開のデータベースに登録する。署名者になり得るユーザjは、公開鍵v_jと秘密鍵v_j(v_j = α^{-s_j} mod p)を生成し、公開鍵を公開のデータベースに登録する。ユーザは複数存在し、署名者も複数存在し得る。

【００５１】Step 1（ユーザ指定と公開）：指定者iは、複数の中からあるユーザ、すなわち署名者jを選び（図1に示す矢印101）、署名者jの公開鍵v_jを乱数r_jを用いて変換したz_jを計算し、z_jに対する署名（Schnorr暗号による署名）を求め（図1に示す手順(1)）、その署名（証明書）を署名者jに安全に送る（図1に示す矢印102）。なお、安全に送る一方法として、前述したE暗号を用いる方法がある。

【００５２】具体的には、指定者iは（秘密の）乱数r_j (r_j ∈ Zq*)を選び、次式により各パラメータを求め、署名(y_j, e_j, z_j)を署名者jに送る。なお、r_j ∈ Zq*は、Zq*からランダムにr_jを選ぶことを表している。

x_j = α^{r_j} mod p
z_j = {v_j}^{r_j} mod p
e_j = h(x_j, z_j)
y_j = r_j + e_j·s_i mod q

【００５３】Step 2（署名の生成）：署名者jは、署名(y_j, e_j, z_j)を手に入れ、次式を確認する。

e_j = h(α^{y_j}·{v_i}^{e_j} mod p, z_j)　　…(1)
z_j =(α^{y_j}·{v_i}^{e_j} mod p)^{-s_k} mod p

【００５４】そして、署名したいメッセージm(j)に対し、次の署名方式で署名を生成する。つまり、署名者jは、秘密の乱数r(j)を選び(r(j) ∈ Zq*)、次式を計算する。

※x(j) = {x_j}^{r(j)} mod p
e(j) = h(x(j), m(j))
y(j) = r(j) + e(j)·s_j mod q

【００５５】そして、((y_j, e_j, z_j), y(j), e(j),

m(j))を署名として、必要な相手に送る（図1に示す手順(3)、矢印103）。

【００５６】Step 3（署名の確認）：上記署名を受信した確認者は最初に、式(1)を確認し、次に次式を確認する（図1に示す手順(4)）。

e(j) = h({x_j}^{y(j)}・{z_j}^{e(j)} mod p, m(j))

【００５７】上記が確認できたならば、受信者（確認者）は、メッセージm(j)に対する署名は指定者iに選ばれた署名者によって生成された署名であると納得する。

【００５８】以上の匿名公開鍵証明書方式には次の性質がある。

(1)指定者iだけが署名者jを指定でき、その証拠が証明書（指定者iの生成したディジタル署名）になる。

(2)指定された署名者jは、証明書と自分の秘密鍵を用いて匿名のままで署名を生成できる。

(3)受信者は、指定者iの公開鍵を用いて上記匿名署名の正当性を確認できる。

(4)匿名署名からその署名者jを判別すること、および、匿名署名を偽造することは困難である。

【００５９】［匿名公開鍵証明書に基づくグループ署名方式］図3は本発明にかかる匿名公開鍵証明書に基づくグループ署名方式(Group Signature based on APKC)を用いたシステムの概念図で、匿名通信ネットワーク(Anonymous Communication Network)を介して、複数のユーザが互いに匿名で通信を行える状況を表している。

【００６０】図3に示す環境には、信頼されたオーソリティZA、ユーザj，k，Vなどがアクセスできる公開データベースが存在し、そこに各ユーザの公開鍵や、共通のパラメータなどが登録されている。公開データベースは、そこに登録された情報をすり替えるような不当な行為を防ぐように、適切に管理されている。また、図3において、あるグループの公開鍵をv_{ZA}とし、そのグループはj，kを含むメンバからなるものとする。

【００６１】同じグループに所属するメンバj，kは、匿名公開鍵証明書(y_j，e_j，z_j)および(y_k，e_k，z_k)＊をそれぞれ、オーソリティZAから発行してもらい、それに基づき、メンバ固有の署名部分(y(j)，e(j)，z(j))を生成し、匿名通信ネットワークを介して、ユーザVにグループ署名((y_j，e_j，z_j)，y(j)，e(j)，z(j))を送る。

【００６２】ユーザVは、グループの公開鍵v_{ZA}を用いて、匿名公開鍵証明書(y_j，e_j,z_j)を検証し、それから計算される値を用いて(y(j)，e(j)，z(j))を検証することにより、グループ署名の正当性を確認することができる。

【００６３】以上の具体的な内容を図4を参考にして説明する。図4は匿名公開鍵証明書に基づくグループ署名方式を説明するための図、図5はグループ署名方式に関する処理の流れの概要を示すフローチャートで、オーソリティZAに指定されたメンバjがグループ署名を生成し、グループ署名を受信した確認者がグループ署名を確認する例を示している。

【００６４】Step 10（準備）：匿名公開鍵証明書方式のStep 0と同じ。

【００６５】Step 11（メンバに対する証明書の発行）：オーソリティZAは、ユーザjをグループのメンバにすることを認め、メンバjの公開鍵を乱数r_jを用いて変換したz_jを計算し、z_jに対する署名（Schnorr暗号による署名）を求め（図4に示す手順(1)）、メンバjに直接送る（図4に示す手順(2)、矢印402）。

【００６６】具体的には、指定者iは（秘密の）乱数r_j (r_j ∈ Zq*)を選び、次式により各パラメータを求め、署名(y_j，e_j，z_j)をメンバjに送る。

x_j = α^{r_j} mod p

z_j = {v_j}^{r_j} mod p

e_j = h(x_j，z_j)

y_j = r_j + e_j・s_{ZA} mod q

【００６７】Step 12（グループ署名の作成）：メンバjは、署名(y_j，e_j，z_j)を手に入れ、次式を確認する。

$$e_j = h(\alpha^{y_j}\cdot\{v_{ZA}\}^{e_j} \bmod p,\ z_j) \quad \cdots(2)$$

$$z_j = (\alpha^{y_j}\cdot\{v_{ZA}\}^{e_j} \bmod p)^{-s_j} \bmod p$$

【００６８】ここで、x_j = α^{y_j}・{v_{ZA}}^{e_j} mod pが成り立っているので、以降は、表記を簡略化するためにx_jを用いる。そして、メンバjは、署名したいメッセージm(j)に対し、次の署名方式でグループ署名を生成する。つまり、メンバjは、秘密の乱数r(j)を選び(r(j) ∈ Zq*)、次式を計算する。

x(j) = {x_j}^{r(j)} mod p

e(j) = h(x(j)，m(j))

y(j) = r(j) + e(j)・s_j mod q

【００６９】そして、((y_j，e_j，z_j)，y(j)，e(j)，m(j))をグループ署名として、必要な相手に送る（図4に示す手順(3)、矢印403）。

【００７０】Step 13（グループ署名の確認）：上記署名※名を受信した確認者は最初に、式(2)を確認し、次に次式を確認する（図4に示す手順(4)）。

e(j) = h({x_j}^{y(j)}・{z_j}^{e(j)} mod p，m(j))

【００７１】上記が確認できたならば、受信者（確認者）は、メッセージm(j)に対する署名はオーソリティZAに認められたメンバによって生成されたグループ署名であると納得する。

【００７２】［グループ署名の開示方法］あるグループ署名を誰が生成したのかを明らかにする、すなわちグループ署名の開示は、次の二つの方法で実現することができる。ただし、対象のグループ署名を((e_k，y_k，z_k)，e(k)，y(k)，m)、その署名者をメンバk、オーソリティZAが生成したz_kに対応する乱数をr_kとする。

【００７３】方法1（信頼されたオーソリティによる開示）：信頼されたオーソリティZAは、証明書を生成する際に用いた乱数r_kを記憶しておくことができるので、あるグループ署名が与えられたとき、その証明書の中のz_kに対応するメンバが誰かを判別することができる。従って、オーソリティZAは、署名者がメンバkで、乱数はr_k(= DL(z_k, v_k))であることがわかる。

【００７４】オーソリティZA（証明者P）は、証明書の公開鍵に対応する秘密鍵と、メンバの公開鍵に対応する秘密鍵とが等しいことを証明する。つまり、前述したDL(v,w) = DL(u, α)を証明する零知識証明プロトコルを利用して、DL(z_k, v_k) = DL(x_k, α)を認証者Vに対して証明する。ただし、x_k = α^{y_k}·v_i^{e_k} mod pである。なお、メンバの公開鍵は公開のデータベースに登録されている。

【００７５】●DL(z_k, v_k) = DL(x_k, α)を証明する零知識証明プロトコル

(1)証明者Pは、v_kを認証者Vに送る。

(2)認証者Vは、乱数aとb(a, b ∈ Zq)を選び、ch = {v_k}^a·α^bを計算し、証明者Pに送る。

(3)証明者Pは、乱数t(t ∈ Zq)を選び、h1 = ch·α^tとh2 = {h1}^{r_k}を計算し、それらを認証者Vに送る。

(4)認証者Vは、(a, b)を証明者Pに送る。

(5)証明者Pは、ch = {v_k}^a·α^bを確認し、tを認証者Vに送る。

(6)認証者Vは、h1 = {v_k}^a·α^{b+t}とh2 = {z_k}^a·{x_k}^{b+t}を確認する。

【００７６】なお、上記のプロトコルの代わりに、同様の証明が可能なプロトコルを使うこともでき、上記のプロトコルだけが利用可能なものというわけではない。

【００７７】方法2（各メンバによる署名の否認）：もう一つの開示方法は、対象のグループ署名が、自分が生成した署名ではないことを、各メンバが証明することによって実現する。

【００７８】メンバj（証明者P）は、そのグループ署名に使われているz_kに対応する秘密鍵が自分の公開鍵に対応する秘密鍵s_j = DL(z_j, x_j)とは異なることを（認証者Vに）示す。つまり、DL(v_j, α) ≠ DL(z_k, x_k)を以下のプロトコルで証明する。ただし、p, q, α, v_i, v_j, z_kおよびx_k = α^{y_k}·{v_i}^{e_k} mod pは、証明者Pと認証者Vの両方が知っていて、両者がパラメータγに合意した上で、以下のプロトコルが実行される。

【００７９】なお、BC(γ, R)は、乱数Rを入力したビットγのビットコミットメントと呼ばれる。これは、あるビットについて'0'か'1'の何れかを選択し、その値が何れかかは、乱数Rが明らかにならない限りわからないような変換値である。従って、乱数Rを示すことでコミットした値を開示でき、同じ変換値に対して違うビットをコミットしたと偽ることが困難であるという性質を

もつ。例えば、I. B.Damgard, "Practical and provably secure release of a secret and exchangeof a signature", Proc. of EUROCRYPT'93, pp.207-217 (1994)に具体例が示されている。

【００８０】●DL(v_j, α) ≠ DL(z_k, x_k)を証明する零知識証明プロトコル

なお、表記を簡略化するために、以下の説明においてはmod pを一部省略する。

(1)認証者Vは、乱数e(e ∈ Zq*)とβ ∈ {0, 1}を選び、(a, b)を次のように計算し、得られた(a,b)を証明者Pに送る。

【００８１】

β = '0' ならば (a, b) = (α^e, {v_j}^e)

β = '1' ならば (a, b) = ({x_k}^e, {z_k}^e)

(2)証明者Pは、A^{-sj} ≡ b (mod p)の成立を確認し、成り立つならばγ = '1' とし、成り立たないならばγ = '0' とし、g = BC(γ, R)を認証者Vに送る。

(3)認証者Vは、eを証明者Pに送る。

(4)証明者Pは、(a, b) = (α^e, {v_j}^e)あるいは(a, b) = ({x_k}^e, {z_k}^e)が成立することを確認し、どちらかが成り立つならばans = Rとし、どちらも成り立たないならばans = stopとし、ansを認証者Vに送る。

(5)認証者Vは、ans = stopの場合はプロトコルの処理を中止し、そうでない場合はBC(γ, R) = gを確認する。

(6)上記(1)から(5)を合計r回繰り返す。

【００８２】なお、上記のプロトコルの代わりに、同様の証明が可能なプロトコルを使うこともでき、上記のプロトコルだけが利用可能なものというわけではない。

【００８３】受信者が開示したいグループ署名を有するときに、以上の二つの方法の何れか片方だけを実行するように制御することに加え、以下のように制御することも可能である。つまり、第一は、二つの方法を任意に選択して実行できるように制御する方法である。第二は、自動的に方法1を実行し、オーソリティZAが有効に機能しない場合は、方法2を実行する制御方法である。第三は、自動的に方法2を実行し、コンタクトできないメンバがあるなどの理由で開示できない場合は、方法1を実行する制御方法である。

【００８４】グループが複数存在する場合は、各グループ（a, b, …とする）に対応する信頼できるオーソリティZA_a, ZA_b, …を設け、各オーソリティZAが固有の公開鍵v_a, v_b, …を管理する、あるいは、信頼できる単一のオーソリティZAが各グループに対応する個別の公開鍵v_a, v_b, …を管理する、あるいは、それらの中間的な方法があり得る。何れの場合にせよ、上記の説明と同様にしてそれぞれのグループ署名を実現できる。

【００８５】以上の実施形態において、指定者あるいは信頼されたオーソリティ、署名者あるいはメンバ、確認者あるいは署名の受信者は、情報処理および通信に関する能力をもつ情報処理装置、例えばパーソナルコンピュ

ータ上において実現可能である。指定者、オーソリティ、署名者、メンバ、確認者、受信者の間における通信は、例えばインターネットのようなネットワーク上で匿名通信を可能にするサービスを運用することによって実現できる。

【００８６】そして、パーソナルコンピュータのような装置により、上記の計算を行い、上記のサービスが利用できる通信ネットワーク上で通信を行うことにより、上記の各ステップを実行して、匿名公開鍵証明書に基づくディジタル署名をグループ署名として用いた情報通信システムを構築することができる。勿論、インターネットに限らず外部と分離された環境、例えば企業内のローカルエリアネットワーク(LAN)においても、情報処理装置を用いて通信ネットワークおよびサービスを構築することにより、匿名公開鍵証明書に基づくディジタル署名をグループ署名として用いた情報通信システムを構築することができる。

【００８７】また、上記では、オーソリティZAが、匿名公開鍵証明書をメンバに直接送る例を説明したが、用途によっては、同証明書を公開データベースに登録し、メンバおよび署名の受信者が公開データベースにアクセスするという方法もあり得る。

【００８８】以上説明したように、本実施形態にかかる匿名公開鍵証明書に基づくディジタル署名をグループ署名として用いた情報通信システムによれば、グループの公開鍵の大きさはグループメンバ数に比例せず、各メンバが、その署名は自分が作成した署名か否かを証明することが可能なグループ署名を実現することができる。従って、公開鍵の大きさがメンバ数に比例しない場合でも、必ずしもオーソリティZAを必要とせず、データサイズの低減と開示の際の融通性を同時に達成でき、コスト低減と使い勝手の向上を達成することができる。

【００８９】また、本実施形態にかかるグループ署名方式は、あるグループ署名に対して、オーソリティZAは、その署名の開示を行うことが可能であり、かつ、各メンバは、その署名は自分が作成した署名か否かを証明することも可能である。従って、オーソリティZAが有効に機能しない場合にも、署名の開示を行うことができ、またその逆も可能であり、より柔軟性のあるシステム、使い勝手のよいシステムにすることができる。

【００９０】また、本実施形態にかかるグループ署名方式は、グループの公開鍵の大きさはグループメンバ数に比例せず、かつ、オーソリティZAは、あるグループ署名に対して、その署名の開示を行うことが可能であり、かつ、各メンバは、その署名は自分が作成した署名か否かを証明することが可能である。従って、データサイズが小さくなる上、オーソリティZAを用いて署名の開示を行うことも、オーソリティZAが有効に機能しない場合に署名の開示を行うこともでき、データサイズの低減と開示の際の融通性の最適化が図れ、コスト低減と使い勝手の最適化を実現することができる。

【００９１】さらに、本実施形態にかかるグループ署名方式は、メンバの新規加入が容易であるという特徴をもつので、メンバの加入の度に処理を繰り返す必要がなく、システム全体の効率を向上することができる。

【００９２】また、前述したPark等が提案するグループ署名方式における公開鍵の大きさと、本発明にかかる方式の公開鍵の大きさとを、素数pを688ビット、一方向性ハッシュ関数のセキュリティパラメータを70ビットに揃えて比較すると、Park等のグループ署名の大きさが1676ビットになる（k'λ = 70とした）のに対して、本発明にかかる方式のグループ署名の大きさは1108ビットである。従って、署名の大きさを約34%削減でき、その分、通信のコストを下げることができる。

【００９３】

【第2実施形態】以下、本発明にかかる第2実施形態の情報通信システムを説明する。なお、第2実施形態において、第1実施形態と略同様の構成については、同一符号を付して、その詳細説明を省略する。

【００９４】前述した実施形態においては、z_jに対する公開鍵証明書をSchnorr暗号の署名を用いて生成し、p, q, x_j, z_jを公開鍵、s_jを秘密鍵として、メッセージm_jに対してSchnorr暗号の署名を適用した。これに対し、第2実施形態においては、メッセージm_jに対して、Schnorr暗号の署名ではなくE暗号(T. E. ElGamal: "Apublic key cryptosystem and a signature scheme based on discretre logarithms, IEEE Transaction on Information Theory, Vol. IT-31, No. 4, pp.469-472, 1985)を適用する。

【００９５】以下では、第1実施形態と異なる点だけを説明する。

【００９６】Step 12（グループ署名の作成）：署名者jは、署名をしたいメッセージm(j)に対し、次の方式で署名を生成する。署名者jは、秘密の乱数k(j)を選び(k(j) ∈Zq*)、次式を計算する。

$$r(j) = \{x_j\}^{k(j)} \bmod p$$

$$s(j) = (m(j) + s_j \cdot r(j)) \cdot k(j)^{-1} \bmod (p - 1)$$

【００９７】そして、((y_j, e_j, z_j), r(j), s(j), m(j))を署名として、必要な相手に送る。

【００９８】Step 13（グループ署名の確認）：上記署名を受信した確認者は最初に、式(2)を確認し、次に次式を確認する。

$$\{x_j\}^{m(j)} \equiv \{z_j\}^{r(j)} \cdot r(j)^{s(j)} \pmod p$$

【００９９】上記が確認できたならば、受信者（確認者）は、メッセージm(j)に対する署名はオーソリティZAに認められたメンバによって生成された署名であると納得する。

【０１００】

【第3実施形態】以下、本発明にかかる第3実施形態の情報通信システムを説明する。なお、第3実施形態におい

て、第1実施形態と略同様の構成については、同一符号を付して、その詳細説明を省略する。

【0101】前述した第1および第2実施形態においては、z_jに対する公開鍵証明書をSchnorr暗号の署名を用いて生成し、メッセージm_jに対してSchnorr暗号またはE暗号を適用する例を説明した。同様に、ディジタル署名標準案としてNIST(NationalInstitute of Standard and Technology)に提案されたDSA（Digital SignatureAlgorthm: 岡本栄司著、暗号理論入門、共立出版、pp.136-138参照）をメッセージm_jに対して用いることも可能である。

【0102】p, q, x_j, z_jを公開鍵、s_jを秘密鍵として用いる。

【0103】Step 12（グループ署名の作成）：署名者jは、署名をしたいメッセージm(j)に対し、次の方式で署名を生成する。署名者jは、秘密の乱数k(j)を選び(k(j)∈Zq*)、次式を計算する。

r(j) = ({x_j}^{k(j)} mod p) mod q

s(j) = k(j)^{-1}·(H(m(j)) - s_j·r(j)) mod q

【0104】そして、(((y_j, e_j), z_j), ((r(j), s(j)), m(j)))を署名として、必要な相手に送る。

【0105】Step 13（グループ署名の確認）：上記署名を受信した確認者は最初に、式(2)を確認し、0 < r(j) < q, 0 < s(j) < q を確認した後、次の処理を行う。

w = (s(j))^{-1} mod q

u1 = H(m(j))·w mod q

u2 = r(j)·w mod q

v = ({x_j}^{u1}·{z_j}^{u2} mod p) mod q

【0106】そして、以上の計算結果からv = r(j)を確認し、確認できたならば受信者はm(j)に対する署名はオーソリティZAに選ばれたメンバによって生成された署名であると納得する。

【0107】

【第4実施形態】以下、本発明にかかる第4実施形態の情報通信システムを説明する。なお、第4実施形態において、第1実施形態と略同様の構成については、同一符号を付して、その詳細説明を省略する。

【0108】前述した第1から第3実施形態においては、z_jに対する公開鍵証明書をSchnorr暗号の署名を用いて生成し、メッセージm_jに対してSchnorr暗号、E暗号またはDSAを適用する例を説明した。これに対して、z_jに対する公開鍵証明書の生成にSchnorr暗号の署名の代わって、E暗号を適用することも可能である。ただし、x_jとz_jが次式の関係を満たすことが保証できるように、署名を適用することに注意する。

【0109】x_j = α^{r_j} mod p

z_j = {v_j}^{r_j} mod p

【0110】具体的には、以下のとおりである。

【0111】Step 11（メンバに対する証明書の発行）：オーソリティZAは、メンバjに対して、秘密の乱数r_jを選び(r_j ∈ Zq*)、次式を計算する。

x_j = α^{r_j} mod p

z_j = {v_j}^{r_j} mod p

S_j = (z_j + s_i·x_j)·{r_j}^{-1} mod (p - 1)

【0112】そして、(x_j, S_j, z_j)を公開鍵証明書とする。

【0113】Step 12（グループ署名の作成）：メンバjは、次式が成り立つことで公開鍵証明書の正当性を確認することができる。

α^{z_j} ≡ {v_i}^{x_j}·{x_j}^{S_j} (mod p)

z_j = {x_j}^{-S_j} mod p

【0114】

【第5実施形態】以下、本発明にかかる第5実施形態の情報通信システムを説明する。なお、第5実施形態において、第1実施形態と略同様の構成については、同一符号を付して、その詳細説明を省略する。

【0115】前述した第4実施形態においては、z_jに対する公開鍵証明書の生成にSchnorr暗号の署名の代わりにE暗号を適用する例を説明した。これに対して、Schnorr暗号やE暗号の代わりに、DSAを適用することも可能である。ただし、x_jとz_jが次式の関係を満たすことが保証できるように、署名を適用することに注意する。

x_j = α^{r_j} mod p

z_j = {v_j}^{r_j} mod p

【0116】具体的には、以下のとおりである。

【0117】Step 11（メンバに対する証明書の発行）：オーソリティZAは、メンバjに対して、秘密の乱数r_jを選び(r_j ∈ Zq*)、次式を計算する。

x_j = α^{r_j} mod p

z_j = {v_j}^{r_j} mod p

S_j = {r_j}^{-1}·(H(z_j) - s_i·x_j) mod q

【0118】そして、(x_j, S_j, z_j)を公開鍵証明書として、メンバjに送る。

【0119】公開鍵証明書の正当性は、次のように確認できる。まず、0 < x_j < p, 0 < S_j < q を確認した後、次の処理を行う。

【0120】Step 12（グループ署名の作成）：メンバjは、次式が成り立つことで公開鍵証明書の正当性を確認することができる。まず、0 < x_j < p, 0 < S_j < q を確認した後、次の処理を行う。

w = {S_j}^{-1} mod q

u1 = H(z_j)·w mod q

u2 = x_j·w mod q

v = α^{u1}·{v_i}^{u2} mod p

【0121】そして、v = x_jを確認する。公開鍵証明書の正当性は、次式が成り立つことで確認できる。

z_j = {x_j}^{-S_j} mod p

【0122】

【他の実施形態】なお、本発明は、複数の機器（例えば

ホストコンピュータ，インタフェイス機器，リーダ，プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機，ファクシミリ装置など）に適用してもよい。

【０１２３】また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読出し実行することによっても、達成されることは言うまでもない。この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。プログラムコードを供給するための記憶媒体としては、例えば、フロッピディスク，ハードディスク，光ディスク，光磁気ディスク，CD-ROM，CD-R，磁気テープ，不揮発性のメモリカード，ROMなどを用いることができる。

【０１２４】また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【０１２５】さらに、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張カードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張カードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【０１２６】本発明を上記記憶媒体に適用する場合、その記憶媒体には、先に説明したフローチャートに対応するプログラムコードを格納することになるが、簡単に説明すると、図6Aおよび6Bのメモリマップ例に示す各モジュールを記憶媒体に格納することになる。すなわち、少なくとも「生成」「確認」および「開示」の各モジュール、または、「底生成」「公開鍵生成」「第一の署名生成」「確認」「第二の署名生成」および「開示」の各モジュールのプログラムコードを記憶媒体に格納すればよい。

【０１２７】

【発明の効果】以上説明したように、本発明によれば、グループの公開鍵の大きさがグループメンバ数に比例する必要がないディジタル署名方式およびそれを用いた情報通信システムを提供することができる。

【０１２８】また、メンバの新規加入が容易であるディジタル署名方式およびそれを用いた情報通信システムを提供することができる。

【０１２９】また、オーソリティが署名の開示を行うことが可能であるとともに、各メンバが、その署名は自分が作成した署名か否かを証明することが可能なディジタル署名方式およびそれを用いた情報通信システムを提供することができる。

【図面の簡単な説明】

【図１】本発明にかかる公開鍵暗号を応用した匿名公開鍵証明書を説明するための図、

【図２】公開鍵証明書に関する処理の流れの概要を示すフローチャート、

【図３】本発明にかかる匿名公開鍵証明書に基づくグループ署名方式を用いたシステムの概念図、

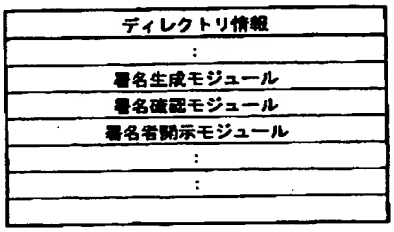
【図４】匿名公開鍵証明書に基づくグループ署名方式を説明するための図、

【図５】グループ署名方式に関する処理の流れの概要を示すフローチャートで、

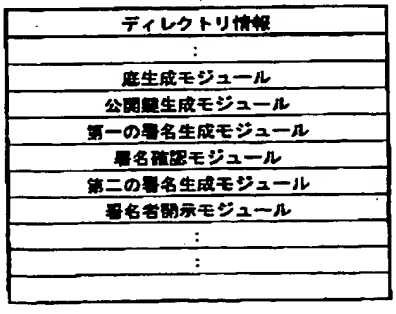
【図６Ａ】本発明にかかるプログラムコードを格納した記憶媒体のメモリマップ例を示す図、

【図６Ｂ】本発明にかかるプログラムコードを格納した記憶媒体のメモリマップ例を示す図である。

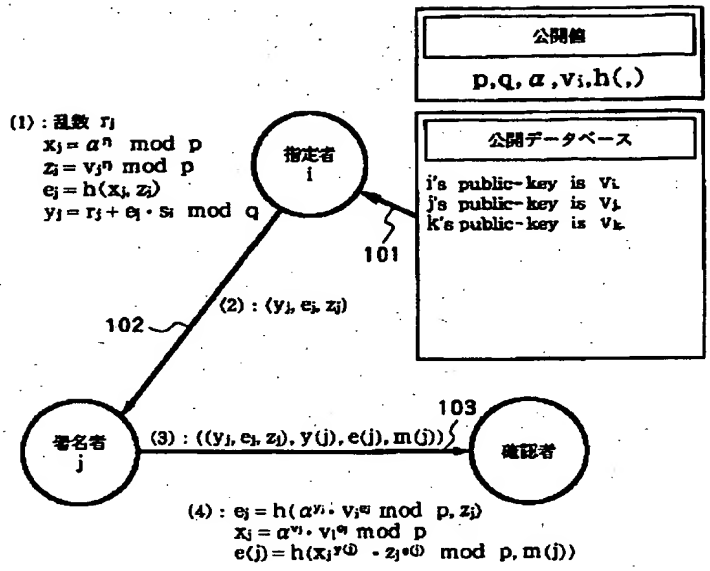
【図6Ａ】



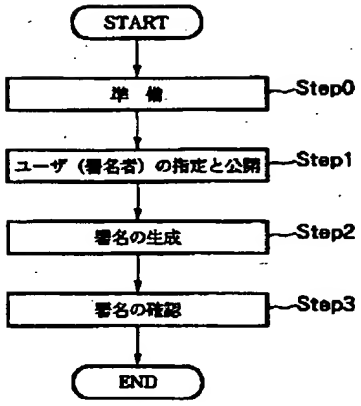
【図6Ｂ】



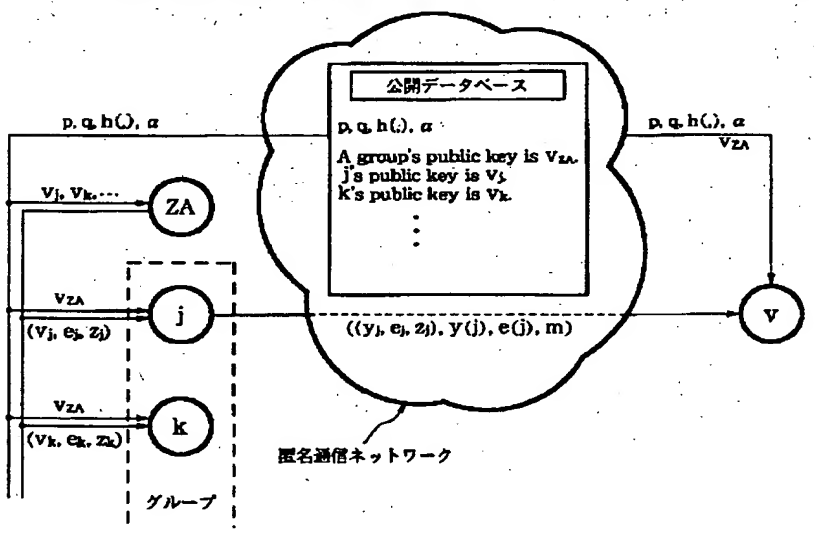
【図1】


公開値
p,q,α,vi,h(,)
公開データベース
i's public-key is vi
j's public-key is vj
k's public-key is vk
指定者 i
署名者 j
確認者
101
102
103
(1)：乱数 rj
xj = α^rj mod p
zj = vj^rj mod p
ej = h(xj, zj)
yj = rj + ej・si mod q
(2)：(yj, ej, zj)
(3)：((yj, ej, zj), y(j), e(j), m(j))
(4)：ej = h(α^yj・vj^ej mod p, zj)
xj = α^yj・vl^ej mod p
e(j) = h(xj^y(j)・zj^e(j) mod p, m(j))

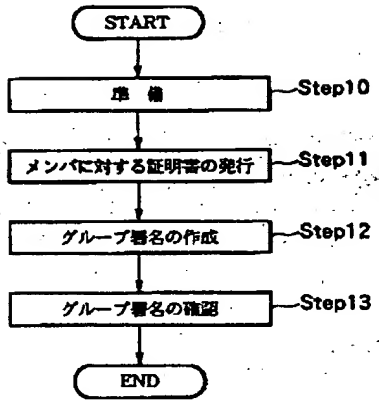
【図2】


START
準 備
Step0
ユーザ（署名者）の指定と公開
Step1
署名の生成
Step2
署名の確認
Step3
END

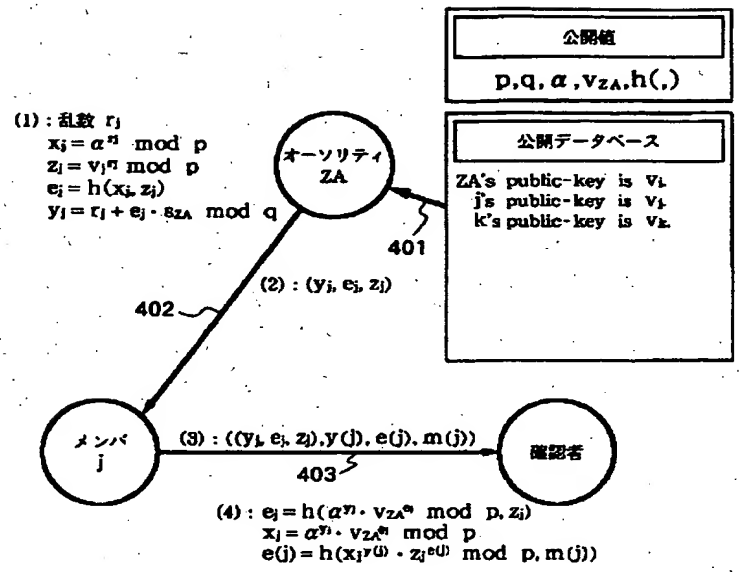
【図3】


公開データベース
p, q, h(,), α
A group's public key is vZA
j's public key is vj
k's public key is vk
p, q, h(,), α
p, q, h(,), α
vZA
vj, vk,…
ZA
vZA
(vj, ej, zj)
j
((yj, ej, zj), y(j), e(j), m)
V
vZA
(vk, ek, zk)
k
匿名通信ネットワーク
グループ

【図5】


START
準 備
Step10
メンバに対する証明書の発行
Step11
グループ署名の作成
Step12
グループ署名の確認
Step13
END

【図４】


公開値
p,q,α,vZA,h(,)
公開データベース
ZA's public-key is vA
j's public-key is vj
k's public-key is vk
オーソリティ
ZA
401
(1) : 乱数 rj
xj = α^rj mod p
zj = vj^rj mod p
ej = h(xj, zj)
yj = rj + ej · sZA mod q
402
(2) : (yj, ej, zj)
メンバ
j
(3) : ((yj, ej, zj),y(j), e(j), m(j))
403
確認者
(4) : ej = h(α^yj · vZA^ej mod p, zj)
xj = α^yj · vZA^ej mod p
e(j) = h(xj^r(j) · zj^e(j) mod p, m(j))

(WPAT)
TI - Digital signature method for information communication - using specifier and signer to provide digital signature which accompanies message sent from signer to verifier
PN - EP-804003 A2 97.10.29 *
PN - JP09298537 A 97.11.18
PN - JP09298536 A 97.11.18
JPNA- J09298537
JPNA- J09298536
PA - (CANO) CANON KK
IC1 - H04L-009/32
AB - EP-804003 A

The digital signature method involves generating public information based on a common public parameter and an uncommon secret parameter. The common public parameter and the public information are converted for use in obtaining a signature for a message.

The signature is generated based on a common public parameter and the uncommon secret parameter. The relationship between the signature and the message is confirmed based on the converted public information and the converted common public parameter.

ADVANTAGE - Does not require special authority to open signature. (Dwg.2/12)

```
?s pn=jp 9298537
      S2       1  PN=JP 9298537
?t s2/5

 2/5/1
DIALOG(R)File 351:DERWENT WPI
(c) 2000 Derwent Info Ltd. All rts. reserv.

011539166    **Image available**
WPI Acc No: 97-515647/199748
XRPX Acc No: N97-428958
 Digital signature method for information communication - using specifier
 and signer to provide digital signature which accompanies message sent
 from signer to verifier
Patent Assignee: CANON KK (CANO  )
Inventor: OISHI K
Number of Countries: 004  Number of Patents: 003
Patent Family:
Patent No  Kind  Date   Applicat No Kind  Date   Main IPC      Week
EP 804003   A2 19971029 EP 97302857  A  19970425 H04L-009/32   199748 B
JP 9298537  A  19971118 JP 96108226  A  19960426 H04L-009/32   199805
JP 9298536  A  19971118 JP 96108225  A  19960426 H04L-009/32   199805

Priority Applications (No Type Date): JP 96108226 A 19960426; JP 96108225 A
  19960426
Cited Patents: No-SR.Pub
Patent Details:
Patent   Kind Lan Pg Filing Notes     Application  Patent
EP 804003   A2 E  30
   Designated States (Regional): DE FR GB
JP 9298537  A     13
JP 9298536  A     11

Abstract (Basic): EP 804003 A
        The digital signature method involves generating public information
    based on a common public parameter and an uncommon secret parameter.
    The common public parameter and the public information are converted
    for use in obtaining a signature for a message.
        The signature is generated based on a common public parameter and
    the uncommon secret parameter. The relationship between the signature
    and the message is confirmed based on the converted public information
    and the converted common public parameter.
        ADVANTAGE - Does not require special authority to open signature.
        Dwg.2/12
Title Terms: DIGITAL; SIGNATURE; METHOD; INFORMATION; COMMUNICATE; DIGITAL;
  SIGNATURE; ACCOMPANIED; MESSAGE; SEND; VERIFICATION
Derwent Class: P85; W01
International Patent Class (Main): H04L-009/32
International Patent Class (Additional): G09C-001/00
File Segment: EPI; EngPI
?logoff
       03may00 09:52:47 User238451 Session D1763.3
       Sub account: P000637
            $6.23    0.283 DialUnits File351
               $7.52  2 Type(s) in Format  5
            $7.52  2 Types
    $13.75  Estimated cost File351
     $0.40  TYMNET
    $14.15  Estimated cost this search
    $14.23  Estimated total session cost   0.433 DialUnits
```